

RTI Recording Service

Getting Started Guide

Version 5.1.0



Your systems. Working as one.



© 2007-2013 Real-Time Innovations, Inc.

All rights reserved.

Printed in U.S.A. First printing.

December 2013.

Trademarks

Real-Time Innovations, RTI, and Connexx are trademarks or registered trademarks of Real-Time Innovations, Inc. All other trademarks used in this document are the property of their respective owners.

Copy and Use Restrictions

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form (including electronic, mechanical, photocopy, and facsimile) without the prior written permission of Real-Time Innovations, Inc. The software described in this document is furnished under and subject to the RTI software license agreement. The software may be used or copied only under the terms of the license agreement.

Third-Party Copyright Notices

Portions of this product include software derived from Fnmatch, (c) 1989, 1993, 1994 The Regents of the University of California. All rights reserved. The Regents and contributors provide this software "as is" without warranty.

Technical Support

Real-Time Innovations, Inc.

232 E. Java Drive

Sunnyvale, CA 94089

Phone: (408) 990-7444

Email: support@rti.com

Website: <https://support.rti.com/>

Contents

1 Welcome to RTI Recording Service

2 Installation

2.1	Downloading Recording Service.....	2-1
2.2	Installing Recording Service	2-2
2.3	Installing the License File	2-3

3 Tutorial—RTI Recording Console

3.1	Setting Up Shapes Demo	3-1
3.2	Recording Data	3-2
3.3	Replaying Data.....	3-4

4 Tutorial—Record and Replay Tools

4.1	Setting Up Shapes Demo	4-1
4.2	Configuring the Record Tool.....	4-3
4.3	Running the Record Tool.....	4-4
4.4	Viewing the Recorded Data with SQLite	4-5
4.5	Using Replay	4-6
4.5.1	Recording Data for Replay.....	4-6
4.5.2	Changing Replay's Configuration File	4-6
4.5.3	Replaying Data at Normal Speed.....	4-8
4.5.4	Replaying Data at a Faster Rate and Using Remote Control	4-8
4.6	Managing the Record Tool Remotely	4-10
4.6.1	Connecting to the Record Tool using Shared Memory	4-11
4.6.2	Modifying the Record Tool While it is Running	4-11

Chapter 1 Welcome to RTI Recording Service

RTI® Recording Service includes these components:

- ❑ *Record*, an *RTI Connex™* (formerly *RTI Data Distribution Service*) application that records both *Connex* discovery and topic data. All recorded data is stored in one or more SQL database files.
- ❑ *Replay*, a tool that can ‘play back’ the recorded data. You even have the option of replaying the data with different data rates or QoS settings.
- ❑ *Recording Console*, a simple graphical user interface (GUI) for using the *Record* and *Replay* tools. This interface significantly reduces *Recording Service* configuration time and complexity, and does not require any programming. The *Recording Console* makes it easy to use *Recording Service* for testing algorithms and other processing logic against pre-recorded test data, conducting regression testing from ‘golden’ data inputs, or recording live data from the field for post-mission analysis.
- ❑ *Convert*, a utility that enables serialized or deserialized data recorded with *Record* to be exported to CSV, HTML, SQL, or XML formats.
- ❑ *sqlite3*, a SQL command-line tool that provides another way to view the data files.

Recording Features:

- ❑ Records data from applications in multiple domains.
- ❑ Records entire Topics, or specific Topic fields, based on POSIX file-name matching expressions.
- ❑ Records all data types except bit-fields.
- ❑ Records to multiple files with configurable file-size limits. Optionally overwrites the oldest file when the maximum number of files has been reached.
- ❑ Records the DDS SampleInfo structure and a timestamp for both discovery data and user data.
- ❑ Records using either Best Effort or Reliable communications.
- ❑ Optionally records data from only specified partitions.
- ❑ Supports remote operation.

Replay Features

- ❑ Publishes data samples that were recorded in serialized format.
- ❑ Highly configurable—you can:
 - Choose which serialized topics to replay
 - Set the replay rate (faster or slower) or use the original rate

- Change the QoS of the publications
- Configure the QoS for the tool itself
- Dynamically control the replay (start, stop, pause) and single-step through the data samples

Chapter 2 Installation

RTI Recording Service is included with *RTI Connexet Messaging*. Use the installation instructions in this chapter only if you are installing *Recording Service* independently (not part of *Connexet Messaging*).

This chapter describes how to download and install *Recording Service*:

- Downloading Recording Service (Section 2.1)
- Installing Recording Service (Section 2.2)
- Installing the License File (Section 2.3)

2.1 Downloading Recording Service

Before you can install *Recording Service*, you must download the distribution file from the RTI Support Portal, accessible from <https://support.rti.com/>. You will need your username and password to log in; these are included in the letter confirming your purchase or evaluation copy. If you do not have this letter, please contact license@rti.com.

Once you have logged into the portal, select the **Downloads** link, then select the distribution that matches the architecture on which you will be running the *Record* tool and the version of your *Connexet* application.

Distribution files follow the naming convention, **RTI_Recording_Service-<version>-<architecture>.exe**. For example, **RTI_Recording_Service-5.x.y-WIN32.exe**.

Save the file to a temporary directory on your computer, such as **/tmp** on UNIX-based systems or **My Documents** on Windows systems. If you need help with the download process, contact support@rti.com.

RTI Shapes Demo: If you want to work through the tutorials in [Chapter 3](#) and [Chapter 4](#), you will also need *RTI Shapes Demo*. It is included with *Connexet Messaging*. It is also available from the RTI Support Portal (<https://support.rti.com/>) or RTI's Downloads page, (<http://www.rti.com/downloads>) in the section titled **Interactive Demonstration**. *Shapes Demo* runs on Windows and Linux systems. Follow the installation instructions that accompany the distribution file.

2.2 Installing Recording Service

On Windows Server 2008 systems: You must have administrator privileges to install *Recording Service*.

On all other Windows systems: Administrator privileges are not required to install *Recording Service*, but you do need write permission to the installation directory. (Note that non-administrator users do *not* have write permission to "Program Files" on Windows Vista and Windows 7 systems).

Note for 64-bit platforms: If you are installing *Recording Service* on an Ubuntu 64-bit platform or a Red Hat Enterprise Linux 5 or 6 64-bit platform, you will also need to install a set of 32-bit libraries *before installing Recording Service*:

- ❑ On Red Hat Enterprise Linux 5 and 6 64-bit platforms, run this command (you will need *root* permission):

```
yum install glibc.i686 libX11.i686 gtk2.i686 libXtst.i686
```

- ❑ On Ubuntu 64-bit platforms, run this command (you will need *root* permission):

```
apt-get install libc6 ia32-libs-multiarch
```

Note: When selecting an installation directory, do not choose one with square brackets ([]) in the path.

To install *Recording Service*, simply run the installation file that you downloaded and follow the on-screen instructions.

After installation, you will end up with the following directory structure under your chosen installation location:

- ❑ **bin/<architecture>/** Executables for *<architecture>*; includes **rtirecconv**, **rtirecord**, **rtirecsh**, **rtireplay**, **rtireplaysh**, and **sqlite3**. You can have several architectures installed.
- ❑ **console/** Files used by *Recording Console*.
- ❑ **doc/** Documentation
- ❑ **examples/** Example configuration files for *Record* and *Replay*
- ❑ **jre/** Java libraries and executables
- ❑ **lib/<architecture>/** Libraries
- ❑ **resource/idl** IDL files used by *Record* and *Replay*
- ❑ **resource/schema** The *Recording Service* XSD and DTD for XML validation and configuration help.
- ❑ **resource/xml** Folder for XML files, including the default *Recording Service* configuration. Please put your own XML files here.
- ❑ **scripts** Scripts to launch the executables.
- ❑ **zip** Files used by *Recording Console*.

Recording Console creates an additional directory:

- ❑ On UNIX-based systems: **<user home directory>/rti/recording_service_5.x.y/console**
- ❑ On Windows systems: **My Documents\RTI\RTI Recording Service 5.x.y**

This directory contains:

- ❑ A **settings.ini** file, which contains operational settings. If you want to reset the application settings, delete this file. A new file will be created the next time you start *Recording Console*.

- A **logs** directory, which may contain message logs in case you need to troubleshoot problems.

2.3 Installing the License File

If your *Recording Service* distribution requires a license file, you will receive one via email from RTI.

If you have *Connext Messaging* and you want to use *RTI Launcher*¹ to start *Recording Service*:

By default, *Launcher* looks for the license file **rti_license.dat** in the top-level directory where you installed *Connext Messaging*. If you choose to save the license file elsewhere, you can configure *Launcher* to look in a different location by using its Configuration tab.

Otherwise:

Save the license file in any location of your choice. When the *Record* tool starts, it will look in these locations until it finds a valid license:

1. The file specified with the **-licenseFile** option when you start the *Record* tool from the command-line.
2. The file specified in the environment variable **RTI_LICENSE_FILE**, which you may set to point to the full path of the license file, including the filename (for example, **C:\RTI\my_rti_license.dat**).
3. The file **rti_license.dat** in the current working directory.
4. The file **rti_license.dat** in the directory specified by the environment variable **NDDSHOME**.

If you have any questions about license installation, please contact support@rti.com.

1. *Launcher* is a convenient GUI-based tool that can start and configure all of your *Connext Messaging* components, including *Recording Service*.

Chapter 3 Tutorial—RTI Recording Console

The best way to learn about *RTI Recording Console* is to start using it. This chapter will walk you through the steps to use the *Console* to record and replay data from *Shapes Demo*.

The tutorial assumes you have the following software installed (using the same version number for both components):

- Recording Service*
- Shapes Demo*

The tutorial walks you through the following exercises:

- [Setting Up Shapes Demo \(Section 3.1\)](#)
- [Recording Data \(Section 3.2\)](#)
- [Replaying Data \(Section 3.3\)](#)

3.1 Setting Up Shapes Demo

We'll use *Shapes Demo* to publish and subscribe to Topics which will appear in the application window as colored moving shapes. Each shape is a Topic; the topic's data includes the shape's color; x and y coordinates; a fillKind (solid, empty, horizontal or vertical stripes); and an angle.

To Begin the Tutorial:

1. Start *two* copies of *Shapes Demo*.

- On Linux systems:

Enter the following command:

```
> <Shapes Demo install directory>/scripts/rtishapesdemo
```

- On Windows systems:

If you have *RTI Launcher*, you can use its Utilities tab to start *Shapes Demo*. Or from the Windows **Start** menu, navigate to **RTI Shapes Demo <version>** and select **RTI Shapes Demo**.

Depending on your security settings, you may see a Windows Security Alert dialog—simply click **Allow Access**.

No configuration changes are necessary. If you choose to use a different domain ID, make sure to use the same value in both instances of *Shapes Demo*, as well as in the *Console*'s Configuration panels.

Repeat, so you are running two instances of *Shapes Demo*.

We'll refer to the first Demo window as the Publisher Demo and the second as the Subscriber Demo.

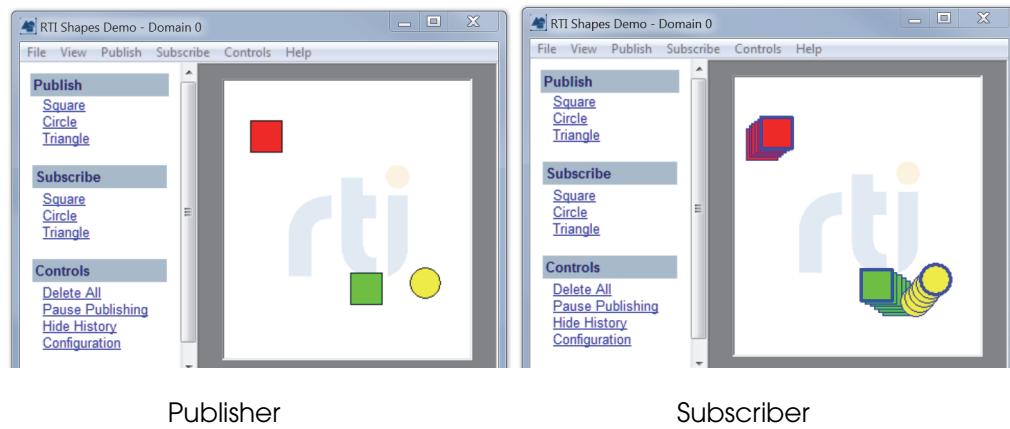
2. In the Publisher Demo, create 2 squares and a circle—all with default QoS settings:
 - a. Under **Publish**, select **Square**.
 - b. In the **Create New Publisher** dialog, select **RED** and click **OK**. (Use the defaults for the other settings).

You should see a red square moving in the display area.
 - c. Use the same process to create a green square and a yellow circle.

You've just created 3 Writers (2 Squares, 1 Circle).
3. In the Subscriber Demo, subscribe to squares and circles:
 - a. Under **Subscribe**, select **Square** and click **OK**. (Leave the default settings).

You've just created a single Reader that will receive data for all squares (all colors).
 - b. Use the same process to create a Reader for circles.

Your two *Shapes Demo* windows should look like this:



By default, Subscribers show the most recent 6 samples—this is why you see extra shapes behind the leading one. The most recent shape has a thick colored border. Older (historical) samples have a thinner border of the same color. Published shapes have a thin black border.

- c. You may minimize these two windows.

3.2 Recording Data

1. Start *Recording Console*.

On UNIX-based systems:

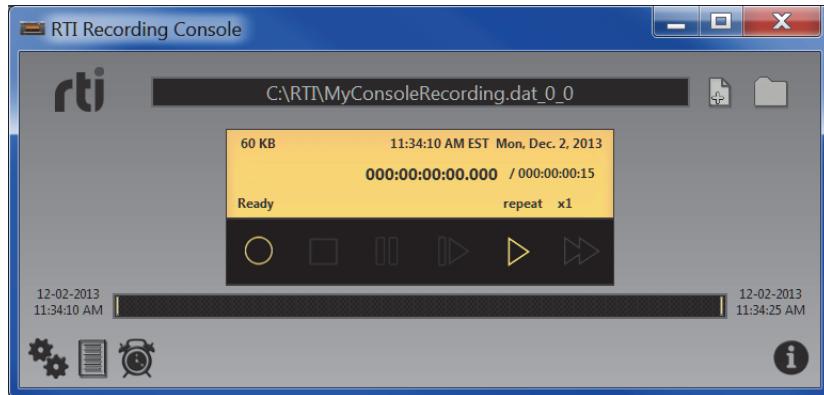
If you have *Connext Messaging*, you can start the *Recording Console* from *RTI Launcher*'s Tools tab. Or open a command prompt, change to the **<Recording Service install directory>/scripts** directory and start the *Console*:

```
> rtirecordingconsole
```

Wait a few seconds for the *Console* to launch.

On Windows systems:

If you have *Connext Messaging*, you can start the *Recording Console* from *RTI Launcher*'s Tools tab. Or from the Start menu, navigate to **RTI Recording Service <version>**, and select **RTI Recording Console**. Wait a few seconds for the *Console* to launch:



2. Create a new file in which to record data from *Shapes Demo*:

- Press the New Recording button  and specify a file name and location. Then click on **Create File**.

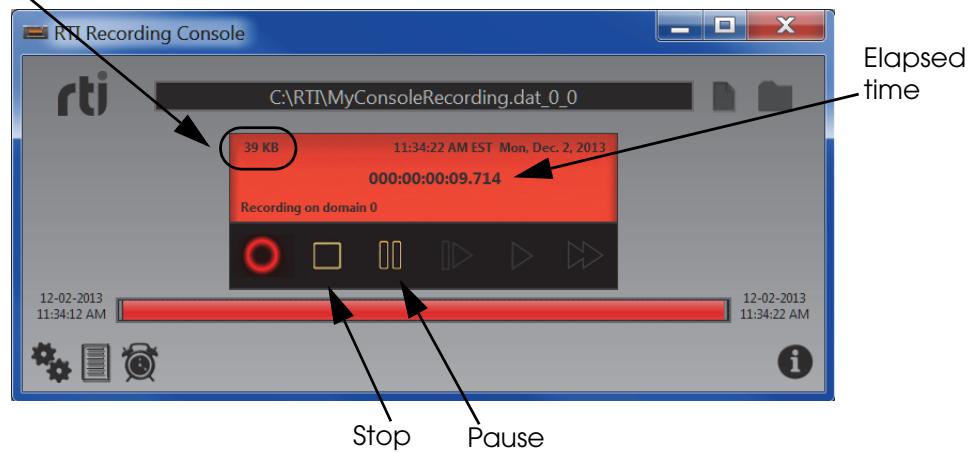
3. Press the Record button to start recording.

(On Windows systems, depending on your security settings, you may see a Windows Security Alert dialog—if so, simply click **Allow Access**.)



4. Wait a few seconds for the *Console* to discover and start recording data from *Shapes Demo*.

File size grows as data is recorded



Tip: If you have trouble recording, see [Troubleshooting Recording Problems \(Section 2.4.2\)](#)

5. In the Publishing Demo, click on the red square and move it around in a pattern, such as a circle or zig-zag. Notice that the red square in the Subscribing Demo is moving the same way. (We will look for this pattern of movement again later in the next section of the tutorial).
6. Press the *Console*'s Stop button .
7. You can close the Publishing Demo. Keep the *Console* and Subscribing Demo running; you will use them in the next section.
8. Notice that the shapes in the Subscribing Demo still appear but they are no longer moving. That's because this copy of *Shapes Demo* is no longer receiving data.

3.3 Replaying Data

In this section, we will replay data that was recorded from *Shapes Demo* in [Section 3.2](#).

This example will replay the recorded data with the same QoS settings as when the data was recorded.

You should still be running *Recording Console* and the Subscribing Demo from [Section 3.2](#).

9. In the *Console*, press the Open Folder button  in the upper-right corner, locate the file whose data is to be replayed, then click **Open**. This will be the same file you recorded in [Section 3.2](#). (If you skipped that part of the tutorial, you can use the example file in `<installation directory>/example/replay`.)

Another way to open a recording file is simply to drag the file from your file browser and drop it into the long black rectangle at the top of the *Console*.

10. Press Play  to begin replaying the data.

In the Subscribing Demo, you will see the objects moving exactly the same as when they were recorded, including when you moved the red square around in a pattern.

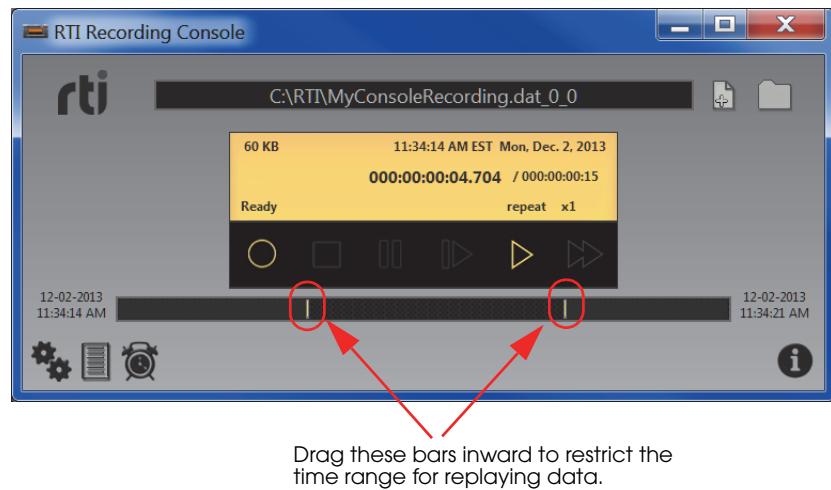


11. Press Stop .

12. Replay the data again, but this time try moving the vertical slider (on the right) which controls the replay rate. Slide it up to go faster, down to go slower. You will be able to see the rate change in the Subscribing Demo. To quickly return to the original speed, press Play .

13. Press Stop .

14. Replay just a section in the middle of the data. Drag the vertical bars on the progress bar inward to shorten the replay time:



15. Press Play  to begin replaying the just the middle section of the data, beginning where you put the left bar, and ending where you put the right bar.



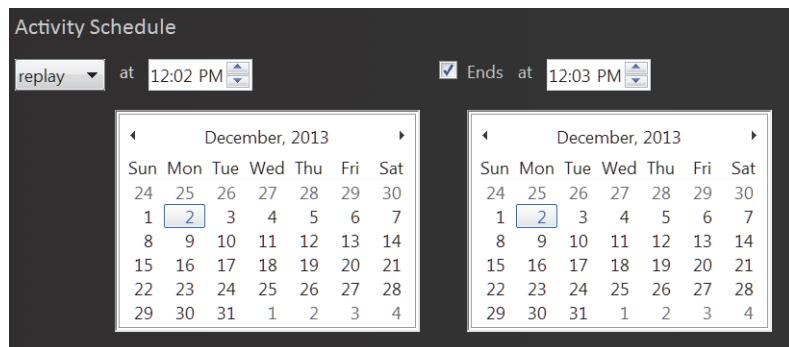
16. Press Stop .

17. Slide the bars back to their original positions (to the far left and far right).

18. Now let's see how to schedule a recording. Press the Schedule button .

19. From the drop-down list on the left, select **replay**.

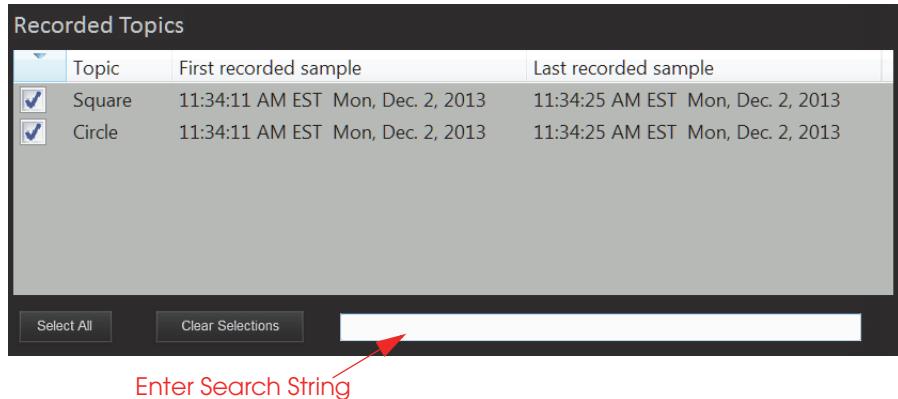
20. Set a starting time for 1 minute from now and a stop time for 1 minute after that. Check the "Ends at" box.



21. Watch the Subscriber Demo and/or the *Console's* display area to confirm that the data is replayed at the scheduled time. If it runs out of data before the stop time it will simply stop early. If you want it to keep repeating the replay, press  to open the Playback Configuration panels and check **Auto repeat** before scheduling the activity.



22. Press the Topics Table button  . This will show you the Topic names, and first and last samples for each Topic.



	Topic	First recorded sample	Last recorded sample
<input checked="" type="checkbox"/>	Square	11:34:11 AM EST Mon, Dec. 2, 2013	11:34:25 AM EST Mon, Dec. 2, 2013
<input checked="" type="checkbox"/>	Circle	11:34:11 AM EST Mon, Dec. 2, 2013	11:34:25 AM EST Mon, Dec. 2, 2013

Enter Search String

23. Press “Clear Selections” and then Play  . Since no topics are selected, no data will be replayed.

24. Enter “Cir” in the search field at the bottom. Now only Circle topics appear in the table. Check the box next to the Circle.

25. Press Play  again. Notice in the Subscriber Demo that only the circle object is being replayed.

This concludes the tutorial for *Recording Console*.

To learn how to use the individual *Replay* and *Record* tools from the command-line (not through the *Console*), see [Chapter 4](#). Or for more information, see the *Recording Service User’s Manual*.

Chapter 4 Tutorial—Record and Replay Tools

While [Chapter 3](#)'s tutorial focused on recording and replaying data from the *Recording Console*, this chapter's tutorial will show you how to use the individual *Replay* and *Record* tools. These are actually the same services that the *Console* invokes.

First, we will generate recordable data via *Shapes Demo*, a *Connext* application. Then we will record data sent between two instances of *Shapes Demo* and view the database tables with the provided SQL tool, `sqlite3`. Next, we will replay the data. Finally, we will learn how to manage the *Record* tool remotely.

Important:

- To understand the contents of each database record, you will need to consult the *RTI Core Libraries and Utilities C API Reference documentation*.
- To use `sqlite3` to view the recorded data, you will need a basic understanding of SQL commands; for information, please visit www.sqlite.org.

The tutorial assumes you have the following software installed (using the same version number for both components):

- Recording Service*
- Shapes Demo*

Exercises in the tutorial include:

- [Setting Up Shapes Demo \(Section 4.1\)](#)
- [Configuring the Record Tool \(Section 4.2\)](#)
- [Running the Record Tool \(Section 4.3\)](#)
- [Viewing the Recorded Data with SQLite \(Section 4.4\)](#)
- [Using Replay \(Section 4.5\)](#)
- [Managing the Record Tool Remotely \(Section 4.6\)](#)

4.1 Setting Up Shapes Demo

We'll use *Shapes Demo* to publish and subscribe to Topics which will appear in the application window as colored moving shapes. Each shape is a Topic; the topic's data includes the shape's color; x and y coordinates; a fillKind (solid, empty, horizontal or vertical stripes); and an angle.

To Begin the Tutorial:

1. Start *two* copies of *Shapes Demo*.

- On Linux systems:

Enter the following command:

```
> <Shapes Demo install directory>/scripts/rtishapesdemo
```

- On Windows systems:

If you have *RTI Launcher*, you can use its Utilities tab to start *Shapes Demo*. Or from the Windows **Start** menu, navigate to **RTI Shapes Demo <version>** and select **RTI Shapes Demo**.

Depending on your security settings, you may see a Windows Security Alert dialog—simply click **Allow Access**.

No configuration changes are necessary. If you choose to use a different domain ID, make sure to use the same value in both instances of *Shapes Demo*, as well as in the *Console's Configuration panels*.

Repeat, so you are running two instances of *Shapes Demo*.

We'll refer to the first Demo window as the Publisher Demo and the second as the Subscriber Demo.

2. In the Publisher Demo, create 2 squares and a circle—all with default QoS settings:
 - a. Under **Publish**, select **Square**.
 - b. In the **Create New Publisher** dialog, select **RED** and click **OK**. (We'll use the defaults for the other settings).

You should see a red square moving in the display area.

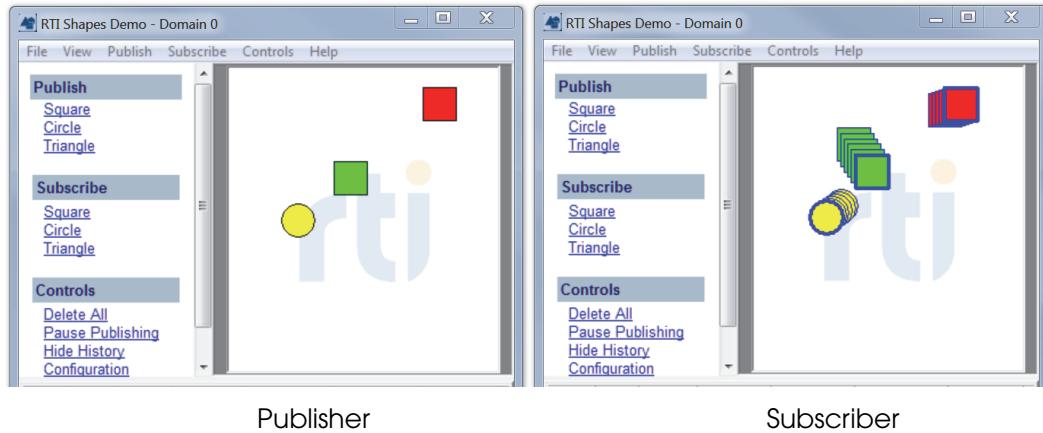
 - c. Use the same process to create a green square and a yellow circle.

You've just created 3 Writers (2 Squares, 1 Circle).
3. In the Subscriber Demo, subscribe to squares and circles:
 - a. Under **Subscribe**, select **Square** and click **OK**. (Leave the default settings.)

You've just created a single Reader that will receive data for all squares (all colors).

 - b. Use the same process to create a Reader for circles.

Your two *Shapes Demo* windows should look like this:



By default, Subscribers show the most recent six samples—this is why you see extra shapes behind the leading one. The most recent shape has a thick colored border. Older (historical) samples have a thinner border of the same color. Published shapes have a thin black border.

- You may minimize these two windows.

4.2 Configuring the Record Tool

You may specify a configuration file when you start the *Record* tool. The configuration file is used to set properties that control what to record, how to record, and where to save the recorded data.

The `<install-dir>/examples/record` directory contains these example files:

- `simple_config.xml`
- `advanced_config.xml`
- `remote_shell.xml`

Let's start learning about the *Record* tool with the simple configuration.

1. Copy `<install-dir>/examples/record/simple_config.xml` to `<install-dir>/scripts/mysimple_record_config.xml`.

On Windows Vista, Windows 7, and Windows 8: If you installed under **Program Files** and do not have write permission to that directory, you will have to put `mysimple_record_config.xml` in your **Documents** folder instead. When you start *Record*, you will need to specify the full path to the file (such as `c:\Users\<user-name>\Documents\mysimple_record_config.xml`).

2. If your computer has an active network interface *and* you are using the default domain ID (0) for *Shapes Demo*: proceed to [Running the Record Tool \(Section 4.3\)](#)—you can use the copied configuration file as is.

Otherwise, open `mysimple_record_config.xml` with an editor of your choice, make the following changes, then save and close the file:

- a. If you changed the domain ID used by *Shapes Demo*, change the `<domain_id>` line to use the same value. For example, if you set the domain ID to 3 in *Shapes Demo*, then change it here too:

Change the bold line to use the same domain ID as Shapes Demo

```
<domain name="domain0">
  <domain_id> 3 </domain_id>
  <deserialize_mode>
    RTIDDS_DESERIALIZemode_ALWAYS
  </deserialize_mode>
</domain>
```

b. If your computer does not have an active network interface, use the following lines in the domain properties section.

Turn off multicast-->

Only use shared memory-->

```
<domain name="domain0">
  <domain_id> 0 </domain_id>
  <participant_qos>
    <discovery>
      <multicast_receive_addresses/>
    </discovery>
    <transport_builtin>
      <mask>SHMEM</mask>
    </transport_builtin>
  </participant_qos>
</domain>
```

4.3 Running the Record Tool

1. Open a command prompt¹ and change to the `<install-dir>/scripts` directory.
2. Start the *Record* tool with your new configuration file (from Section 4.2). For example (if you saved the configuration file someplace other than the `scripts` directory, specify the full path to the file):

```
> rtirecord -cfgFile mysimple_record_config.xml -cfgName example
```

On Windows systems when using an evaluation version of *Recording Service*: If you are not logged in as Administrator, the license manager will display a message:

"Error in saving configuration settings: You are not a license administrator."

This message is benign and the *Record* tool should run normally.

On Windows systems, depending on your security settings, you may see a Windows Security Alert dialog—if so, simply click **Unblock**.

3. Wait about one minute to give the *Record* tool time to record data from *Shapes Demo*.
4. Stop the *Record* tool by pressing **<Control-C>**. Leave the command prompt window open—you will use it in Section 4.4.

1. On Windows systems: from the **Start** menu, select **Accessories, Command Prompt**.

4.4 Viewing the Recorded Data with SQLite

1. After running and stopping the *Record* tool as described in Section 4.3, list the contents of the current directory. You will see that the *Record* tool created a file called **simple_config.dat_0_0**. (The first 0 is a fileset number, the second 0 is a segment number.)
2. From the same command prompt used to run the *Record* tool, open the database file with `<Recording Service install directory>/bin/<architecture>/sqlite3` to view the recorded data. For example, assuming your command prompt is in `<Recording Service install directory>/scripts` (where the data files are), enter the following (replace `<architecture>` to match your installation):

```
> .../bin/<architecture>/sqlite3 simple_config.dat_0_0
```

To use `sqlite3`, you will need some basic knowledge of SQL commands—for information, please visit www.sqlite.org.

3. Start by displaying a list of the tables in the database:

```
sqlite> .tables
```

The *Record* tool creates this table for discovery data:

- **DCPSPublication** (data type DDS_PublicationBuiltInTopicData)

This table stores the recorded topic data:

- **Circle\$RecordAll\$domain0**
- **Square\$RecordAll\$domain0**

Topic table names follow this format: *TopicName\$RecordGroupName\$DomainName*. A “RecordGroup” is a way to specify what topics to record from what domains. This concept is explained in the *Recording Service User’s Manual*. In this example, the RecordGroup is called **RecordAll**.

The *Record* tool also creates two tables for internal use, **RTILog** and **RTIVersion**.

4. SQLite provides a number of builtin commands. Start with `.help`:

```
sqlite> .help
```

5. It is useful to turn column headings on:

```
sqlite> .headers on
```

6. Query the tables using standard SQL syntax. (Note that using “limit x” syntax at the end of each select statement will limit the number of records displayed. Without a limit, you will likely see too many records scrolling off your display.) Here are some examples:

To list 3 records of data on Topic Circle:

```
sqlite> select * from Circle$RecordAll$domain0 limit 3;
```

To get the largest x-coordinate value for a Circle:

```
sqlite> select max(x) from Circle$RecordAll$domain0;
```

To get 10 samples from the table, in descending order by the column named “SampleInfo_reception_timestamp”:

1. For some platforms such as Ubuntu and SUSE, you will not find a folder for your specific architecture. You can run `sqlite3` from any `<architecture>` folder.

```
sqlite> select * from Circle$RecordAll$domain0 order by
    SampleInfo_reception_timestamp desc limit 10;
```

To find the number of samples recorded:

```
sqlite> select count(*) from Circle$RecordAll$domain0;
```

To find all x values greater than 100:

```
sqlite> select x from Circle$RecordAll$domain0 where x > 100;
```

To find all x, y values, where x is greater than y:

```
sqlite> select x,y from Circle$RecordAll$domain0 where x > y;
```

Try other SQL commands to explore the recorded data.

When you are ready to exit sqlite3, enter:

```
sqlite> .exit
```

4.5 Using Replay

In this section, you will learn about:

- Recording Data for Replay (Section 4.5.1)
- Changing Replay's Configuration File (Section 4.5.2)
- Replaying Data at Normal Speed (Section 4.5.3)
- Replaying Data at a Faster Rate and Using Remote Control (Section 4.5.4)

4.5.1 Recording Data for Replay

To get started with *Replay*, you can either (a) use the example database provided in `examples/replay/replay_database.dat_0_0` or (b) configure and run *Record* using the steps in [Section 4.2](#) and [Section 4.3](#) with the `replay_example` configuration instead of `example`:

```
> rtirecord -cfgFile mysimple_record_config.xml -cfgName replay_example
```

Remember the name of the data file that gets created; you will need it in the next section.

Replay can replay data that has been recorded in either serialized or deserialized form. For files recorded in serialized mode, SQLite is unable to look at the individual fields in the sample data. For files recorded in deserialized mode, it is important for the *Record tool* to record *all* of the fields of the sample data, as *Replay* is unable to replay partial data.

4.5.2 Changing Replay's Configuration File

Replay requires an XML configuration file. We will use the example file, `simple_config.xml`, provided in `<install-dir>/examples/replay`. (Note that this is not the same file that you changed in [Section 4.5.1](#)—that was *Record*'s `simple_config.xml`; now we are using *Replay*'s `simple_config.xml`.)

1. Copy `<install-dir>/examples/replay/simple_config.xml` to `<install-dir>/scripts/mysimple_replay_config.xml`.

On Windows Vista, Windows 7, and Windows 8: If you installed under **Program Files** and do not have write permission to that directory, you will have to put `mysimple_replay_config.xml` in your **Documents** folder instead. When you start *Replay*, you will need to specify the full path to the file (such as `c:\Users\<user-name>\Documents\mysimple_replay_config.xml`).

2. The configuration file assumes that the database containing serialized samples is named `replay_database.dat_0_0` (this file is in `<install directory>/examples/replay`); if you want to replay this example database, copy it to `<install directory>/script` and skip the rest of this step.

However, if you want to replay a file that has a different name, you will need to edit `mysimple_replay_config.xml`. To change the name of the database file in `mysimple_replay_config.xml`, change the value of the `<filename>` parameter (just below the tag for `<replay_database>`).

The lines to look for are:

```
<!-- Source Database, Required -->
<replay_database name="simple_config">
  <filename>replay_database.dat_0_0</filename>
```

Note: There are three configurations in `simple_config.xml`, so you will need to change the filename in three places if you want to use a different filename.

3. The configuration file assumes you want to use domain ID 0 for the participant that will replay the data and domain ID 1 for remote administration. If that's alright, you can skip this step.

Otherwise, you will need to change the domain ID(s) in the configuration file, or specify the replay domain ID and/or the remote administration domain ID on the command line with `-domainIdBase` and `-remoteAdministrationDomainId`, respectively.

In the configuration file, the domain ID assignment for replaying the data is right after the filename parameter and looks like this:

```
<participant>
  <domain_id>0</domain_id>
<participant_qos>
```

The replay participant domain ID assignment occurs in three places, so be sure to change all of them.

The domain ID assignment for remote administration is near the top of the service configuration) and looks like this:

```
<administration>
  <name>example</name>
  <domain_id>1</domain_id>
</administration>
```

The remote administration domain ID is only set in one of the examples.

4.5.3 Replaying Data at Normal Speed

This example will replay the recorded data once, at the same speed and with the same QoS as when the data was recorded. The data will be replayed on the domain ID specified in the configuration (default 0), or on the domain ID specified on the command line.

1. Start one copy of *Shapes Demo*, configured as in [Section 4.2](#), and create subscribers for Squares and Circles.
2. Open a command prompt¹ and change to the `<install-dir>/scripts` directory.
3. Start *Replay* with your new configuration file (from [Section 4.5.2](#)). For example (if you saved the configuration file someplace other than the **scripts** directory, specify the full path to the file):

```
> rtireplay -cfgFile mysimple_replay_config.xml -cfgName simple_example
```

Expected Results:

In *Shapes Demo*, you should see the objects moving exactly the same as when they were recorded.

4. When the example has finished replaying, press **<Control-C>** to exit.
5. In the XML configuration file, look at the configuration named **simple_example** (the first example).

This example defines a single database (**simple_config**), with a single session (**A_Session**), and a single topic (**All_Topic**), yet *Replay* replayed the data from both Squares and Circles. This happened because the topic description uses wildcards ("*") to indicate that *Replay* should replay any topics in the database with any values of topic name, type name, record group name, and domain name. *Replay* created a separate DataWriter for each topic that it found matching the selection parameters; in this example, there were two: Squares, and Circles. In the next example we will be more discriminating.

4.5.4 Replaying Data at a Faster Rate and Using Remote Control

This example demonstrates a higher degree of control over the parameters used during replay. We will replay the data faster than its original speed by using a different configuration named **fast_replay**. And while the data is replaying, you will see how to remotely control the replay by pausing, resuming, and single-stepping.

1. Open a command prompt² and change to the `<install-dir>/scripts` directory.
2. Start *Replay* in the same manner described in [Section 4.5.3](#), with these minor changes to the command line:
 - Replace `-cfgName simple_example` with `-cfgName fast_replay`
 - Add the parameter `-remoteAdministrationDomainId` followed by the domain ID that you want to use for remote operation. (You can omit this option if you already changed the remote administration domain ID assignment in the configuration file as discussed in [Section 4.5.2](#)).

For example (if you saved the configuration file someplace other than the **scripts** directory, specify the full path to the file):

1. On Windows systems: from the **Start** menu, select **Accessories, Command Prompt**.
2. On Windows systems, from the **Start** menu, select **Accessories, Command Prompt**.

```
> rtireplay -cfgFile mysimple_replay_config.xml
  -cfgName fast_replay
  -remoteAdministrationDomainId <other_domain_id>
```

Expected Results:

You should see the objects moving in *Shapes Demo* at a higher rate of speed than in the previous example.

In the XML configuration file, look at the configuration named **fast_replay** (the second example). You will see that the higher rate of replay was configured using the `<rate>` parameter of the `<replay_service> <time_control>` section of this configuration.

By this time you should also notice that this replay does not stop, but repeats the movement of the objects once the original data has been replayed. You can see in the configuration that this was selected by setting the mode **LOOP** for the `<start_mode>` parameters of the `<replay_topic> <topic_time_control>` sections of **A_Topic** and **B_Topic**. (Repeating the data allows us to experiment with remote control, without having the data stream come to an end.)

You can also see that both **A_Topic** and **B_Topic** are more discriminating in their choice of topics to replay, explicitly naming the topics and types of interest. Separating the two topics allows them to be controlled individually.

3. Now we'll see how to control *Replay* remotely using a program called **rtireplaysh** (the *Replay* shell).

- a. Open a command prompt¹ and change to the `<install-dir>/scripts` directory.
- b. Start *Replay Shell* using the same domain ID that you used for the **-remoteAdministrationDomainId** argument in the previous step:

```
> rtireplaysh -domainId <other_domain_id>
```

You will see this prompt:

```
"Replay Shell> "
```

- c. Enter "help" to see what commands are supported.

Replay Shell commands use this format:

```
<command> <replay_service> <entity> [value]
```

where:

- **<command>** is one of the supported commands.
- **<replay_service>** is the name given to the *Replay* service by one of the following, in descending order of precedence:
 - Value specified with the **-appName** command-line option used when starting *Replay* (*highest precedence*)
 - Value for the `<replay_service><administration><name>` element
 - Value for the `<replay_service>` name attribute (*lowest precedence*)
- **<entity>** is any one of the service entities expressed in this hierarchical form: `<database-name>[::<session-name>[::<topic-name>]]`.
- **value** depends on the command. Not all commands require a value.

Note: *In this release, not all commands are supported for all entity levels.* Please see the *Recording Service Release Notes* for details on which modes are currently supported.

1. On Windows systems, from the **Start** menu, select **Accessories, Command Prompt**.

The database-name must match a name from a `<replay_database>` tag in the configuration file that you specified when starting *Replay*, such as:

```
<replay_database name="simple_config">
```

Similarly, if you specify a session-name, it must match a name from a `<session>` tag within the specified database, such as:

```
<session name="A_Session">
```

If you specify a topic-name, it must match a name from a `<replay_topic>` tag within the specified session, such as:

```
<replay_topic name="All_Topic">
```

4. While topics are being replayed, you can pause or stop the replay. For example, to pause the replay, use this command:

```
Replay Shell> pause example simple_config::A_Session
```

Note: The *Replay Shell* has no line-editing capabilities, so take extra care when entering commands.

5. Once paused, you can resume or step like this:

```
Replay Shell> step example simple_config::A_Session
Replay Shell> resume example simple_config::A_Session
```

6. It is also possible to control individual topics:

```
Replay Shell> pause example simple_config::A_Session::A_Topic
Replay Shell> step example simple_config::A_Session::A_Topic
Replay Shell> resume example simple_config::A_Session::A_Topic
Replay Shell> rate example simple_config::A_Session::A_Topic 1
Replay Shell> rate example simple_config::A_Session::B_Topic 1
```

4.6 Managing the Record Tool Remotely

The *Record* tool can be managed remotely using one of two ways:

- RTI Record Shell*
- Another *Connexxt* application using special message types

See Chapter 7 in the *Recording Service User's Manual* for details.

This section contains examples of how to use the *Record Shell* commands:

- [Connecting to the Record Tool using Shared Memory \(Section 4.6.1\)](#)
- [Modifying the Record Tool While it is Running \(Section 4.6.2\)](#)

The configuration files used in the examples are in `<install directory>/examples/record`.

Note: If you are using different domain IDs for running the *Record* tool, you must edit the configuration files to use your domain IDs. By default, the example configuration files use domain IDs 0 and 1.

4.6.1 Connecting to the Record Tool using Shared Memory

1. Copy `<install directory>/examples/record/advanced_config.xml` to `<install directory>/scripts/advanced_config.xml`
2. Change to this directory: `<installation directory>/scripts`
3. Start the *Record* tool as follows:
`./rtirecord -cfgFile advanced_config.xml -cfgName remote_example`
4. In another shell, change to this directory: `<installation directory>/scripts`
5. In the shell from step 4, start the *Record Shell*:
`./rtirecsh -domainId 0 -noUdpv4`
6. Enter **info** (in the shell that is running `rtirecsh`) to see which *Record* session it is connected to.
7. Enter **status** to check *Record*'s status. Note that State is RECORDING.
8. Enter **stop** to tell *Record* to stop recording.
9. Enter **status**. The State value should be STOPPED, indicating that recording has been stopped.
10. Reconfigure the *Record* tool with a file that is local to the *Record* tool:
`configure advanced_example -remotefile
 .../examples/record/advanced_config.xml`
11. Wait a few seconds and enter **status** in the shell. The *Record* tool should now be configured based on **advanced_config.xml**.
12. Wait a few seconds and enter **stop** in the shell. The *Record* tool should now be stopped.
13. Reconfigure the *Record* tool with a file that is local to the *Record Shell* tool:
`configure advanced_example -localfile
 .../examples/record/advanced_config.xml`
14. Wait a few seconds and enter **status** in the shell. The *Record* tool should now be configured based on **advanced_config.xml**.
15. Enter the **stop** command to stop recording.
16. Enter the **shutdown** command to shut down the *Record* tool .
17. Enter the **exit** command to exit the shell.

4.6.2 Modifying the Record Tool While it is Running

1. Copy `<installation directory>/examples/record/remote_shell.xml` to `<installation directory>/scripts/remote_shell.xml`.
2. Start the *Record* tool with the configuration file, `remote_shell.xml`, in the `examples` directory.
`./rtirecord -cfgFile remote_shell.xml -cfgName remote_shell`
3. Start *Shapes Demo* and create publishers for a Square, Circle and Triangle.
4. Start the remote shell.
`./rtirecsh -domainId 1`
5. In the shell, enter:
`model`

You will see the following results:

```
Current Record Model:
=====
<recorder>
  <model name="configuration">
    <domain name="domain0" state="RUNNING">
      <domain_id>0</domain_id>
    </domain>
    <domain name="domain1" state="RUNNING">
      <domain_id>1</domain_id>
    </domain>
    <topic_group name="Square" state="RUNNING" ref_count="1">
      <topics>
        <topic_expr>Square</topic_expr>
        <exemption></exemption>
      </topics>
      <field_expr>*</field_expr>
    </topic_group>
    <topic_group name="Triangle" state="RUNNING" ref_count="1">
      <topics>
        <topic_expr>Triangle</topic_expr>
        <exemption></exemption>
      </topics>
      <field_expr>*</field_expr>
    </topic_group>
    <record_group name="SquareTriangle" state="RUNNING" ref_count="1">
      <domain_ref>
        <element>domain0</element>
      </domain_ref>
      <topic_ref>
        <element>Square</element>
        <element>Triangle</element>
      </topic_ref>
    </record_group>
  </model>
</recorder>
```

6. In the shell, enter the following command (all on one line):

```
add <dds><recorder><topic_group name="Circle"><topics><topic_expr>
Circle</topic_expr></topics><field_expr>*</field_expr>
</topic_group></recorder></dds>
```

The new model:

```
<recorder>
  <model name="configuration">
    <domain name="domain0" state="RUNNING">
      <domain_id>0</domain_id>
    </domain>
    <domain name="domain1" state="RUNNING">
      <domain_id>1</domain_id>
    </domain>
    <topic_group name="Circle" state="RUNNING" ref_count="0">
      <topics>
        <topic_expr>Circle</topic_expr>
        <exemption></exemption>
      </topics>
      <field_expr>*</field_expr>
    </topic_group>
    <topic_group name="Square" state="RUNNING" ref_count="1">
```

```

<topics>
    <topic_expr>Square</topic_expr>
    <exemption></exemption>
</topics>
<field_expr>*</field_expr>
</topic_group>
<topic_group name="Triangle" state="RUNNING" ref_count="1">
    <topics>
        <topic_expr>Triangle</topic_expr>
        <exemption></exemption>
    </topics>
    <field_expr>*</field_expr>
</topic_group>
<record_group name="SquareTriangle" state="RUNNING" ref_count="1">
    <domain_ref>
        <element>domain0</element>
    </domain_ref>
    <topic_ref>
        <element>Square</element>
        <element>Triangle</element>
    </topic_ref>
</record_group>
</model>
</recorder>

```

Notice that the topic group we just added has a **ref_count** of 0. This means no record groups are referencing it at the moment. Now let's fix that.

7. In the shell, enter the following command (all on one line):

```

add <dds><recorder><record_group name="Circle"><domain_ref>
    <element>domain0</element></domain_ref><topic_ref><element>
    Circle</element></topic_ref></record_group></recorder></dds>

```

New model:

```

<recorder>
    <model name="configuration">
        <domain name="domain0" state="RUNNING">
            <domain_id>0</domain_id>
        </domain>
        <domain name="domain1" state="RUNNING">
            <domain_id>1</domain_id>
        </domain>
        <topic_group name="Circle" state="RUNNING" ref_count="1">
            <topics>
                <topic_expr>Circle</topic_expr>
                <exemption></exemption>
            </topics>
            <field_expr>*</field_expr>
        </topic_group>
        <topic_group name="Square" state="RUNNING" ref_count="1">
            <topics>
                <topic_expr>Square</topic_expr>
                <exemption></exemption>
            </topics>
            <field_expr>*</field_expr>
        </topic_group>
        <topic_group name="Triangle" state="RUNNING" ref_count="1">
            <topics>
                <topic_expr>Triangle</topic_expr>
                <exemption></exemption>
            </topics>
            <field_expr>*</field_expr>

```

```
</topic_group>
<record_group name="Circle" state="RUNNING" ref_count="1">
    <domain_ref>
        <element>domain0</element>
    </domain_ref>
    <topic_ref>
        <element>Circle</element>
    </topic_ref>
</record_group>
<record_group name="SquareTriangle" state="RUNNING"
ref_count="1">
    <domain_ref>
        <element>domain0</element>
    </domain_ref>
    <topic_ref>
        <element>Square</element>
        <element>Triangle</element>
    </topic_ref>
</record_group>
</model>
</recorder>
```

Now we're recording the topic "Circle". You can open the database to verify this.